# Measure energy consumption using machine learning

Team members: Swastina S , Vasundhra J, Surya L, Vimala Devi S,Sathya P, Sowmiya S.

**Phase 4 :** Development part 2

**Project title:** Measure energy consumption

**Introduction :** Energy consumption models are used to determine energy requirements based on input parameters. They can be used to:

Determine energy supply requirements

Determine consumer consumption variations when technology is added or upgraded

Energy consumption models are made up of two components:

A static component related to equipment with constant power consumption

A dynamic component related to energy consumption in power amplifiers, attributed to traffic load.

**Data set**

DateTime Count

10/01/2004 - 01/10/2005 2,425

01/10/2005 - 04/21/2005 2,425

04/21/2005 - 07/31/2005 2,426

07/31/2005 - 11/09/2005 2,425

11/09/2005 - 02/18/2006 2,426

02/18/2006 - 05/30/2006 2,425

05/30/2006 - 09/08/2006 2,426

09/08/2006 - 12/18/2006 2,425

12/18/2006 - 03/29/2007 2,425

03/29/2007 - 07/08/2007 2,425

07/08/2007 - 10/17/2007 2,426

| | |
|---|---|
| 10/17/2007 - 01/26/2008 | 2,425 |
| 01/26/2008 - 05/07/2008 | 2,425 |
| 05/07/2008 - 08/16/2008 | 2,426 |
| 08/16/2008 - 11/25/2008 | 2,425 |
| 11/25/2008 - 03/06/2009 | 2,426 |
| 03/06/2009 - 06/15/2009 | 2,425 |
| 06/15/2009 - 09/24/2009 | 2,426 |
| 09/24/2009 - 01/03/2010 | 2,425 |
| 01/03/2010 - 04/14/2010 | 2,424 |
| 04/14/2010 - 07/24/2010 | 2,426 |
| 07/24/2010 - 11/02/2010 | 2,426 |
| 11/02/2010 - 02/11/2011 | 2,424 |
| 02/11/2011 - 05/23/2011 | 2,425 |
| 05/23/2011 - 09/02/2011 | 2,426 |
| 09/02/2011 - 12/12/2011 | 2,425 |
| 12/12/2011 - 03/22/2012 | 2,425 |
| 03/22/2012 - 07/01/2012 | 2,426 |
| 07/01/2012 - 10/10/2012 | 2,426 |
| 10/10/2012 - 01/19/2013 | 2,423 |
| 01/19/2013 - 04/30/2013 | 2,425 |
| 04/30/2013 - 08/09/2013 | 2,426 |
| 08/09/2013 - 11/18/2013 | 2,425 |
| 11/18/2013 - 02/27/2014 | 2,426 |
| 02/27/2014 - 06/08/2014 | 2,424 |
| 06/08/2014 - 09/17/2014 | 2,426 |
| 09/17/2014 - 12/27/2014 | 2,427 |

12/27/2014 - 04/08/2015 2,425

04/08/2015 - 07/18/2015 2,426

07/18/2015 - 10/27/2015 2,425

10/27/2015 - 02/05/2016 2,427

02/05/2016 - 05/16/2016 2,425

05/16/2016 - 08/25/2016 2,426

08/25/2016 - 12/04/2016 2,427

12/04/2016 - 03/15/2017 2,425

03/15/2017 - 06/24/2017 2,426

06/24/2017 - 10/03/2017 2,426

10/03/2017 - 01/12/2018 2,427

01/12/2018 - 04/23/2018 2,425

04/23/2018 - 08/03/2018 2,426

1Oct04

3Aug18

Label Count

9581.00 - 9903.28 86

9903.28 - 10225.56 284

10225.56 - 10547.84 610

10547.84 - 10870.12 959

10870.12 - 11192.40 1,663

11192.40 - 11514.68 2,162

11514.68 - 11836.96 2,811

11836.96 - 12159.24 3,177

12159.24 - 12481.52 3,648

12481.52 - 12803.80 3,892

| | |
|---|---|
| 12803.80 - 13126.08 | 4,112 |
| 13126.08 - 13448.36 | 4,293 |
| 13448.36 - 13770.64 | 4,858 |
| 13770.64 - 14092.92 | 5,389 |
| 14092.92 - 14415.20 | 5,969 |
| 14415.20 - 14737.48 | 5,889 |
| 14737.48 - 15059.76 | 6,031 |
| 15059.76 - 15382.04 | 6,241 |
| 15382.04 - 15704.32 | 6,210 |
| 15704.32 - 16026.60 | 5,715 |
| 16026.60 - 16348.88 | 5,313 |
| 16348.88 - 16671.16 | 4,767 |
| 16671.16 - 16993.44 | 4,196 |
| 16993.44 - 17315.72 | 4,128 |
| 17315.72 - 17638.00 | 3,834 |
| 17638.00 - 17960.28 | 3,453 |
| 17960.28 - 18282.56 | 3,133 |
| 18282.56 - 18604.84 | 2,797 |
| 18604.84 - 18927.12 | 2,515 |
| 18927.12 - 19249.40 | 2,290 |
| 19249.40 - 19571.68 | 1,977 |
| 19571.68 - 19893.96 | 1,705 |
| 19893.96 - 20216.24 | 1,555 |
| 20216.24 - 20538.52 | 1,160 |
| 20538.52 - 20860.80 | 999 |
| 20860.80 - 21183.08 | 872 |

21183.08 - 21505.36 626

21505.36 - 21827.64 482

21827.64 - 22149.92 409

22149.92 - 22472.20 320

22472.20 - 22794.48 249

22794.48 - 23116.76 174

23116.76 - 23439.04 115

23439.04 - 23761.32 94

23761.32 - 24083.60 41

24083.60 - 24405.88 25

24405.88 - 24728.16 29

24728.16 - 25050.44 12

25050.44 - 25372.72 3

25372.72 - 25695.00 1

9.58k

25.7k

2004-12-31 01:00:00

13478.0

2004-12-31 02:00:00

12865.0

2004-12-31 03:00:00

12577.0

2004-12-31 04:00:00

12517.0

2004-12-31 05:00:00

12670.0

2004-12-31 06:00:00

13038.0

2004-12-31 07:00:00

13692.0

2004-12-31 08:00:00

14297.0

2004-12-31 09:00:00

14719.0

2004-12-31 10:00:00

14941.0

2004-12-31 11:00:00

15184.0

2004-12-31 12:00:00

15009.0

2004-12-31 13:00:00

14808.0

2004-12-31 14:00:00

14522.0

2004-12-31 15:00:00

14349.0

2004-12-31 16:00:00

14107.0

2004-12-31 17:00:00

14410.0

2004-12-31 18:00:00

15174.0

2004-12-31 19:00:00

15261.0

2004-12-31 20:00:00

14774.0

2004-12-31 21:00:00

14363.0

2004-12-31 22:00:00

14045.0

2004-12-31 23:00:00

13478.0

2005-01-01 00:00:00

12892.0

2004-12-30 01:00:00

14097.0

2004-12-30 02:00:00

13667.0

2004-12-30 03:00:00

13451.0

2004-12-30 04:00:00

13379.0

2004-12-30 05:00:00

13506.0

2004-12-30 06:00:00

14121.0

2004-12-30 07:00:00

15066.0

2004-12-30 08:00:00

15771.0

2004-12-30 09:00:00

16047.0

2004-12-30 10:00:00

16245.0

2004-12-30 11:00:00

16377.0

2004-12-30 12:00:00

16138.0

2004-12-30 13:00:00

15886.0

2004-12-30 14:00:00

15503.0

2004-12-30 15:00:00

15206.0

2004-12-30 16:00:00

15049.0

2004-12-30 17:00:00

15161.0

2004-12-30 18:00:00

16085.0

2004-12-30 19:00:00

16508.0

2004-12-30 20:00:00

16306.0

2004-12-30 21:00:00

16223.0

2004-12-30 22:00:00

15931.0

2004-12-30 23:00:00

15207.0

2004-12-31 00:00:00

14316.0

2004-12-29 01:00:00

15223.0

2004-12-29 02:00:00

14731.0

2004-12-29 03:00:00

14503.0

2004-12-29 04:00:00

14432.0

2004-12-29 05:00:00

14531.0

2004-12-29 06:00:00

15087.0

2004-12-29 07:00:00

16018.0

2004-12-29 08:00:00

16968.0

2004-12-29 09:00:00

17149.0

2004-12-29 10:00:00

17261.0

2004-12-29 11:00:00

17194.0

2004-12-29 12:00:00

17011.0

2004-12-29 13:00:00

16749.0

2004-12-29 14:00:00

16546.0

2004-12-29 15:00:00

16403.0

2004-12-29 16:00:00

16161.0

2004-12-29 17:00:00

16380.0

2004-12-29 18:00:00

17330.0

2004-12-29 19:00:00

17738.0

2004-12-29 20:00:00

17468.0

2004-12-29 21:00:00

17093.0

2004-12-29 22:00:00

16736.0

2004-12-29 23:00:00

15984.0

2004-12-30 00:00:00

14900.0

2004-12-28 01:00:00

17580.0

2004-12-28 02:00:00

17158.0

2004-12-28 03:00:00

17002.0

2004-12-28 04:00:00

16923.0

2004-12-28 05:00:00

17191.0

2004-12-28 06:00:00

17908.0

2004-12-28 07:00:00

18944.0

2004-12-28 08:00:00

19752.0

2004-12-28 09:00:00

19882.0

2004-12-28 10:00:00

19544.0

2004-12-28 11:00:00

19309.0

2004-12-28 12:00:00

18756.0

2004-12-28 13:00:00

18201.0

2004-12-28 14:00:00

17666.0

2004-12-28 15:00:00

17203.0

2004-12-28 16:00:00

16935.0

2004-12-28 17:00:00

17207.0

2004-12-28 18:00:00

18349.0

2004-12-28 19:00:00

18815.0

2004-12-28 20:00:00

18599.0

2004-12-28 21:00:00

18480.0

2004-12-28 22:00:00

18135.0

2004-12-28 23:00:00

17112.0

2004-12-29 00:00:00

16047.0

2004-12-27 01:00:00

16718.0

2004-12-27 02:00:00

16150.0

2004-12-27 03:00:00

16090.0

2004-12-27 04:00:00

16223

**Overview of the process**

**Prepare the data** This includes cleaning data , removing outliers the handling missing values.

**Perform feature selection** This can be done using a variety of methods such as correlation analysis information gain and recursive feature elimination.

**Train the model** There are many different machine learning algorithms that can be used for measure energy consumption . Some popular choices include linear regression random forest and gradient boosting machines.

**Evaluate with the model** This can be done by calculating the main square error or the root means square error of the models prediction on held out test set.

**Deploy the model** Once the model has been evolveted and found to be performing well it can be deployed to protection so that it can be used to predicts the energy consumption for an hour.

**PROCEDURE**

**Feature selection**is

**1) Identify the target variable:** This is the variable that you want to predict such as energy consumption.

**2) Explore the data :** This will help you to understand the relationships between the different features and the target variable you can use data visualisation under correlation analysis to identify features that are highly correlated with the target variable.

**3) Remove redundant features:** If two features are highly correlated with each other then you can remove one of the fishes as they are likely to contain redundant information.

**4) Remove in irrelevant h features:** If feature is not call later with the target variable then you can remove it , as it is unlikely to be useful for prediction.

**Feature selection**

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import warnings


warnings.filterwarnings("ignore", category=UserWarning)


from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.svm import SVR

from sklearn.metrics import mean_squared_error, r2_score


RED = "\033[91m"

GREEN = "\033[92m"

YELLOW = "\033[93m"

BLUE = "\033[94m"

RESET = "\033[0m"


df = pd.read_csv("/kaggle/input/hourly-energy-consumption/AEP_hourly.csv")


df["Datetime"] = pd.to_datetime(df["Datetime"])
```

DATA CLEANING

```
print(BLUE + "\nDATA CLEANING" + RESET)
```

Check for missing values

missing_values = df.isnull().sum()

print(GREEN + "Missing Values : " + RESET)

print(missing_values)

Handle missing values

df.dropna(inplace=True)

Check for duplicate values

duplicate_values = df.duplicated().sum()

print(GREEN + "Duplicate Values : " + RESET)

print(duplicate_values)

Drop duplicate values

df.drop_duplicates(inplace=True)

DATA ANALYSIS

print(BLUE + "\nDATA ANALYSIS" + RESET)

Summary Statistics

summary_stats = df.describe()

print(GREEN + "Summary Statistics : " + RESET)

print(summary_stats)

SUPPORT VECTOR MODELLLING

print(BLUE + "\nMODELLING" + RESET)

Reduce the dataset size for faster training

df = df.sample(frac=0.2, random_state=42)

Split the data into features (Datetime) and target (AEP_MW)

X = df[["Datetime"]]

```
y = df["AEP_MW"]

Split the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(

    X, y, test_size=0.2, random_state=42

)

Preprocess the features (Datetime) to extract the day of the year

X_train["DayOfYear"] = X_train["Datetime"].dt.dayofyear

X_test["DayOfYear"] = X_test["Datetime"].dt.dayofyear

Convert X_train and X_test to NumPy arrays

X_train = X_train["DayOfYear"].values.reshape(-1, 1)

X_test = X_test["DayOfYear"].values.reshape(-1, 1)

Standardize the data

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)

Create an SVR (Support Vector Regression) model with a linear kernel

svr = SVR(kernel="linear", C=1.0)

Train the SVR model

svr.fit(X_train_scaled, y_train)

Predict on the test set

y_pred = svr.predict(X_test_scaled)

Evaluate the model

mse = mean_squared_error(y_test, y_pred)

r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")

print(f"R-squared: {r2}")
```

Plot the actual vs. predicted values

plt.figure(figsize=(10, 6))

plt.scatter(X_test, y_test, color="b", label="Actual")

plt.scatter(X_test, y_pred, color="r", label="Predicted")

plt.xlabel("Day of the Year")

plt.ylabel("Energy Consumption (MW)")

plt.title("SVR Model: Actual vs. Predicted")

plt.legend()

plt.grid()

plt.show()

DATA VISUALIZATION

print(BLUE + "\nDATA VISUALIZATION" + RESET)

Line plot

print(GREEN + "LinePlot : " + RESET)

plt.figure(figsize=(10, 6))

sns.lineplot(data=df, x="Datetime", y="AEP_MW")

plt.xlabel("Datetime")

plt.ylabel("Energy Consumption (MW)")

plt.title("Energy Consumption Over Year")

plt.grid()

plt.show()

Histogram

print(GREEN + "Histogram : " + RESET)

plt.figure(figsize=(10, 6))

```
plt.hist(
    df["AEP_MW"],
    bins=100,
    histtype="barstacked",
    edgecolor="white",
)
plt.xlabel("AEPMW")
plt.ylabel("Frequency")
plt.title("Histogram of MEGAWATT USAGE")
plt.show()
```

DATA CLEANING

Missing Values :

```
Datetime    0
AEP_MW      0
dtype: int64
```

Duplicate Values :

0

DATA ANALYSIS

Summary Statistics :

```
                        Datetime        AEP_MW
count                     121273  121273.000000
mean   2011-09-02 03:17:01.553025024   15499.513717
min          2004-10-01 01:00:00    9581.000000
25%          2008-03-17 15:00:00   13630.000000
```

| | | |
|---|---|---|
| 50% | 2011-09-02 04:00:00 | 15310.000000 |
| 75% | 2015-02-16 17:00:00 | 17200.000000 |
| max | 2018-08-03 00:00:00 | 25695.000000 |
| std | NaN | 2591.399065 |

MODELLING

Mean Squared Error: 6758395.805638685

R-squared: 0.00270160624748228

**Model training:**

Model training is a process of teaching a machine learning model to measure energy consumption. It involve feeding the model historical data.

Once the model is strained it can be used to measure energy consumption for a new data for example you could use the model to measure energy consumption.

**Prepare the data** This involves cleaning the data removing any errors are inconsistencies and transforming the data into a format that is compatible with the machine learning algorithm that you will be using.

**Split the data into training and test sets** The training set will be used to change the model on the test set will be used to evaluate the performance and the model of an unseen data.

**Choose the machine learning algorithm** Data cleaning , data analysis , Support vector modelling, Data visualisation .

**Turn the hyper parameters of the algorithm** The hyperparameters of an a machine learning algorithm aur parameters that control the learning process. It is important to tune the hyper parameters of the algorithm to optimise its performance.

**Train the model on training set:** This involves feeding the training data to the model allowing its to learn the relationships between features and energy consumption.

**Evolved the model on test set:** This involves feeding the test test data to the model measuring how will it predicts the measure energy consumption. If the model performs well well on the test sets then you can be confident that it will generalize well to new data.

**Model evaluation:**

Model evaluation is the process of assessing the performance of a machine learning model on unseen data. This is important to ensure that the model will generalize well to new data .

There are a number of different metrics that can be used to evaluate the performance of a house price prediction model .some of the most common metrics include :

**Mean squared error (MSE) :** This metric measures the average squared difference between the measured and actual energy consumption.

**Root mean squared error (RMSE):** This metrics is the square root of the MSE.

**R squared:** This metric measures how well the model explains the variation in the actual energy consumption. In addition to these metrics, it is also important to consider the following factors when when evaluating a energy consumption model.

Bias: Bias is the tendency of a model to consistently over or underestimate energy consumption.

Variance: Variance is the measure of how much the protections of a model very around the true energy consumed.

Interpretability : Interpretability is the ability to understand how the model makes its protections. This is important for energy consumption models as it allows users to understand the factors.

**<u>Feature engineering:</u>**

Feature engineering is is a crucial aspect of building a energy consumption using machine learning. Was creating a new features transforming existing ones and selecting the most relevant variables to improve the models predictive power here or some feature engineering ideas for energy consumption measurement.

**<u>Various feature to perform model training:</u>**

- **Use a variety of feature engineering techniques**

Feature engineering is the process of transforming raq data into features that more informative and a protective for a machine learning models.

- **Use cross validation**

Cross validation is a technique for evolving the performance of machine learning model on unseen data. It is important to use cross validation to evaluate the performance of your model during a training process. This will help you to avoid over fitting and to Ensure that you are model will be generalize will to new data.

- **Use ensemble methods**

Ensemble methods are a machine learning methods that combine the protections of a multiple models to produce or a more accurate protection ensemble methods can often are so better performance then I entireual machine learning models.

- **Hold out test sets**

A hold out test set is a set of data that is not used to train or evaluate the model during the training process. This data is used to evolve the performance of the model on unseen data after training process is complete.

- **Compare the model to a baseline**

A baseline is a simple model that is used to compare the performance of your model to. For example you could use the mean value as a baseline.

- **Analyze the models predictions**

Once you have a little evaluated the performance of the model you can analyse the models prediction to identify any patterns or biosis. This will help you to understand and weaknesses of the model and two improve it.

**Conclusion :**

In the quest to build an accurate and Reliable energy consumption measurement model we have embarked on a journey that encompasses critical phases from feature selection to model training and evaluation. Each of the stages place and dispensable role in crafting a model that can provide meaningful insights and estimates for one of the most significant financial decision make real estate transactions.